

DETAILED ACTION

1. This office action is in response to the communication filed on 06/24/2009.
2. Claims 1-20 are pending.

Response to Arguments

3. Applicant's arguments regarding the 35 USC 112, first paragraph rejection have been withdrawn due to the amendments.
4. Applicant's arguments regarding the 35 USC 103(a) rejection have been fully considered but are moot in view of new ground(s) of rejection.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Multer et al. (US 2002/0040369, hereafter Multer), in view of Song et al. (US 2003/0065947, hereafter Song), further in view of Vert et al. (US 2001/0008019,

hereafter Vert) and Chasman et al. (US 2007/0180075, hereafter Chasman), and Sudharshana et al. (OTA mobile device software management, hereafter Sudharshana), in view of Noble (US 2004/0064591)

7. For claim 1, Multer discloses a method for remotely configuring at least one mobile device comprising:

storing at a server least one configuration parameter relating to a software application (fig. 6-8, fig. 12, server storing relational hierarchy containing devices and associated applications on the devices and items related to settings associated with each application in [0136], [0242], or configuration parameters, see further, fig. 10 and 11, an application such as Outlook has associated items or application settings such as notes, emails, tasks, calendar, bookmarks... with multiple data fields settings in fig. 13), wherein each of the configuration parameters specifies configuration information as a function of the application (fig. 10-12, [0136], each item is an individual entity or record of application's specific data, e.g., an application such as Outlook has associated items such as notes, emails, tasks, calendar, bookmarks... with multiple data fields settings);

receiving at the server a synchronization request from a mobile device participating in a particular application ([0222], sync triggering from a device by a particular application), the synchronization request to include a device identification (ID) ([0223], device name ID)

determining an associated configuration parameter for the mobile device as a function of the particular application (fig. 12, [0241], [0242], each application is associated with items or configuration parameters);

determining an associated device profile for the mobile device as a function of the device ID ([0223], device type by device ID containing device name and class)

determine whether to configure the mobile device using the determined associated configuration parameter (fig. 12, [0220], determine whether synchronization of application items is necessary);

initiating a process to configure the mobile device using a generated application resource (fig. 15, [0223], determine device profile, then [0225]-[0227], resolving conflicts in data by finding data deltas and applying the data deltas to appropriate application items).

Multer does not disclose storing, at the server, in a descriptor file a registry for each of the at least one mobile device; and querying, in the descriptor file, a registry associated with the mobile device.

However, Song discloses storing and accessing a database of registry of each mobile device (fig. 3, 4, each device has a registry record at the registry server)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer and Song to maintain a device specific registry record at the server so that device specific information can be retrieved separately.

Multer-Song does not disclose each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device.

However, Vert discloses each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device (fig. 4, a registry for a machine with installed programs 1-n and associated program configuration parameters such as name and data fields on the right)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song and Vert to maintain a device specific registry record associated with all installed applications at the server so that device specific application information can be retrieved separately.

Multer-Song-Vert does not disclose the server is a middleware server.

However, Chasman discloses the same (fig. 1, fig. 2, application server with a master database for storing business object type or application and configuration information, application server is read as an middleware server, see Yu et al. US 2007/0226155, [0100], an application server is a middleware server)

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song, Vert and Chasman to implement the synchronization scheme of Multer to a dedicated server such as a middleware server to improve system performance.

Multer-Song-Vert-Chasman does not disclose: retrieving, from an application resources database an application resource to be deployed to the mobile device using a generated resource identifier (ID), and initiating a process to configure the mobile device using the retrieved application resource.

However, Sudharshana discloses:

generating a resource identifier from both the associated device profile and the associated configuration parameter (3.9, 3.11, generating a resource patch ID using device ID and configuration parameter—software version);

retrieving, from an application resources database, an application resource to be deployed to the mobile device using the generated resource ID and initiating a process to configure the mobile device using the retrieved application resource (3.9, 3.11, 3.13, generated patch download from management server to mobile device)

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song, Vert and Chasman and Sudharshana to implement simplify management of patches or resources to be deployed to mobile devices by generating ID's for each patch.

Multer-Song-Vert-Chasman-Sudharshana does not disclose wherein the generated resource ID is generated from both the associated device profile and the associated configuration parameter. Sudharshana teaches generating a unique ID of a resource (patch) associated device profile and the associated configuration parameter (3.11 first par., ME model and e.g., software version).

However, Noble discloses a method of generating a unique ID based on some parameters ([0009], claim 6, generating a unique ID based on a given network ID and a device ID)

It would have been obvious to one skilled in the art at the time of the invention to combine the teachings of Multer-Song-Vert-Chasman-Sudharshana and Noble to generate a resource ID uniquely from given parameters such as device profile (model) and associated configuration parameter (software version). The motivation would be to uniquely distinguish each generated ID with one another efficiently and to backwardly retrieve the parameters from the ID if needed.

8. For claim 13, Multer discloses a program storage device including instructions for remotely configuring at least one mobile device comprising:

storing at a server at least one configuration parameter for each the at least one mobile device, wherein each of the configuration parameters specifies configuration information as a function of a device profile (fig. 12, hierarchy containing devices and associated applications on the devices and associated items or configuration parameters, fig. 6-8, fig. 12, server storing relational hierarchy containing devices and associated applications on the devices and items related to settings associated with each application in [0136], [0242], or configuration parameters, see further, fig. 10 and 11, an application such as Outlook has associated items or application settings such as notes, emails, tasks, calendar, bookmarks... with multiple data fields settings in fig. 13),

receiving at the server a synchronization request from a mobile device participating in a particular application ([0222], sync triggering from a device by a particular application), the synchronization request to include a device identification (ID) ([0223], device name ID);

determining an associated configuration parameter for the mobile device as a function of the particular application (fig. 12, [0241], hierarchy containing application relations with associated items or configuration parameters);

determining an associated device profile for the mobile device as a function of the device ID ([0223], user device profile by device ID)

determine whether to configure the mobile device using the determined associated configuration parameter (fig. 12, [0220], determine whether synchronization of application items is necessary);

if the determined associated configuration parameter is not located ([0046], differencing including determining whether data does not currently exist), initiating a process to configure the mobile device using a generated application resource (fig. 15, [0223], determine device profile, then [0225]-[0227], resolving conflicts in data by finding data deltas and applying the data deltas to appropriate application items).

Multer does not disclose storing, at the server, in a descriptor file a registry for each of the at least one mobile device; and querying, in the descriptor file, a registry associated with the mobile device.

However, Song discloses storing and accessing a database of registry of each mobile device (fig. 3, 4, each device has a registry record at the registry server)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer and Song to maintain a device specific registry record at the server so that device specific information can be retrieved separately.

Multer-Song does not disclose each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device;

However, Vert discloses each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device (fig. 4, a registry for a machine with installed programs 1-n and associated program configuration parameters such as name and data fields on the right)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song and Vert to maintain a device specific registry record associated with all installed applications at the server so that device specific application information can be retrieved separately.

Multer-Song-Vert does not disclose the server is a middleware server.

However, Chasman discloses the same (fig. 1, fig. 2, application server with a master database for storing business object type or application and configuration information, application server is read as an middleware server, see Yu et al. US 2007/0226155, [0100], an application server is a middleware server)

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer and Chasman to implement the synchronization scheme of Multer to a dedicated server such as a middleware server to improve system performance.

Multer-Song-Vert-Chasman does not disclose: retrieving, from an application resources database an application resource to be deployed to the mobile device using a generated resource identifier (ID); and initiating a process to configure the mobile device using the retrieved application resource.

However, Sudharshana discloses:

retrieving, from an application resources database, an application resource to be deployed to the mobile device using the generated resource ID and initiating a process to configure the mobile device using the retrieved application resource (3.9, 3.11, 3.13, generated patch download from management server to mobile device)

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song, Vert and Chasman and Sudharshana to implement simplify management of patches or resources to be deployed to mobile devices by generating ID's for each patch.

Multer-Song-Vert-Chasman-Sudharshana does not disclose wherein the generated resource ID is generated from both the associated device profile and the associated configuration parameter. Sudharshana teaches generating a unique ID of a resource (patch) associated device profile and the associated configuration parameter (3.11 first par., ME model and e.g., software version).

However, Noble discloses a method of generating a unique ID based on some parameters ([0009], claim 6, generating a unique ID based on a given network ID and a device ID)

It would have been obvious to one skilled in the art at the time of the invention to combine the teachings of Multer-Song-Vert-Chasman-Sudharshana and Noble to generate a resource ID uniquely from given parameters such as device profile (model) and associated configuration parameter (software version). The motivation would be to uniquely distinguish each generated ID with one another efficiently and to backwardly retrieve the parameters from the ID if needed.

9. For claim 8, Multer discloses a system for remotely configuring at least one mobile device comprising:

a server, wherein the server stores device specific configuration files configuration parameters, the configuration parameters associating a network application with at least one configuration option (fig. 12, hierarchy containing devices and associated applications on the devices and associated items or configuration parameters, fig. 6-8, fig. 12, server storing relational hierarchy containing devices and associated applications on the devices and items related to settings associated with each application in [0136], [0242], or configuration parameters, see further, fig. 10 and 11, an application such as Outlook has associated items or application settings such as notes, emails, tasks, calendar, bookmarks... with multiple data fields settings in fig. 13),

a configuration module (fig. 6, [0018], [0019], difference engine), wherein the configuration module:

receives a configuration request from a particular mobile device ([0222], sync triggering from a device by a particular application);

determines associated device specific configuration files appropriate for the particular mobile device ([0223], user device profile by device ID);

determine whether to the associated device specific configuration files are already installed on the mobile device (fig. 12, [0220], determine whether synchronization of application items is necessary);

a deployment module initiates a deployment of the associated device specific configuration files for the particular mobile device (fig. 15, [0223], determine device profile, then [0225]-[0227], resolving conflicts in data by finding data deltas and applying the data deltas to appropriate application items)

Multer does not disclose a descriptor file stored in the server to store a registry for each of the at least one mobile device; and querying, in the descriptor file, a registry associated with the particular mobile device.

However, Song discloses storing and accessing a database of registry of each mobile device (fig. 3, 4, each device has a registry record at the registry server)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer and Song to maintain a device specific registry record at the server so that device specific information can be retrieved separately.

Multer-Song does not disclose each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device;

However, Vert discloses each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device (fig. 4, a registry for a machine with installed programs 1-n and associated program configuration parameters such as name and data fields on the right)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song and Vert to maintain a device specific registry record associated with all installed applications at the server so that device specific application information can be retrieved separately.

Multer-Song-Vert does not disclose the server is a middleware server.

However, Chasman discloses the same (fig. 1, fig. 2, application server with a master database for storing business object type or application and configuration information, application server is read as an middleware server, see Yu et al. US 2007/0226155, [0100], an application server is a middleware server)

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song, Vert and Chasman to implement the synchronization scheme of Multer to a dedicated server such as a middleware server to improve system performance.

Multer-Song-Vert-Chasman does not disclose: retrieving, from an application resources database the associated device specific configuration files using the generated resource ID; and initiating a process to configure the mobile device using the retrieved application resource.

However, Sudharshana discloses:

retrieving, from an application resources database the associated device specific configuration files using the generated resource ID; and initiating a process to configure the mobile device using the retrieved application resource (3.9, 3.11, 3.13, generated patch download from management server to mobile device).

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song, Vert and Chasman and Sudharshana to implement simplify management of patches or resources to be deployed to mobile devices by generating ID's for each patch.

Multer-Song-Vert-Chasman-Sudharshana does not disclose generating a resource ID from both the associated device profile and the associated configuration parameter. Sudharshana teaches generating a unique ID of a resource (patch) associated device profile and the associated configuration parameter (3.11 first par., ME model and e.g., software version).

However, Noble discloses a method of generating a unique ID based on some parameters ([0009], claim 6, generating a unique ID based on a given network ID and a device ID)

It would have been obvious to one skilled in the art at the time of the invention to combine the teachings of Multer-Song-Vert-Chasman-Sudharshana and Noble to generate a resource ID uniquely from given parameters such as device profile (model) and associated configuration parameter (software version). The motivation would be to uniquely distinguish each generated ID with one another efficiently and to backwardly retrieve the parameters from the ID if needed.

10. For claim 9, Multer-Song-Vert-Chasman-Sudharshana further discloses the network application includes the collective behavior of a plurality of network devices and each mobile device is associated with a device profile including at least one of a hardware profile and a software profile (Multer, fig. 12, software profile, Chasman, fig. 4 step 430, update message regarding a sales opportunity application software type from the client device and collected from a user).

11. For claim 10, Multer-Song-Vert-Chasman-Sudharshana further discloses determining associated device specific configuration files further includes determining a particular network application associated with the particular mobile device (Multer, [0222], Chasman, fig. 4, update request with application type field associated with identifier); determining configuration options for the particular mobile device participating in the network application using the configuration parameters (Multer, fig. 12); determining the device specific configuration files as a function of the device profile for the particular mobile device (Multer, fig. 12, Chasman, [0024] lines 3-14, comparing identifiers in the master database with identifiers in the update request to synchronize configuration data (in this case the status change) to the latest version).

12. For claims 2 and 14, the claims are rejected as in claims 1 and 13. Multer-Song-Vert-Chasman-Sudharshana further discloses initiating a process to configure the mobile device as a function of the associated configuration parameter includes transmitting device specific configuration files to the mobile device as a function of the associated configuration parameter (Multer, fig. 15, 16, get delta, resolve conflict then apply delta to device, Chasman, [0024] lines 14-19, synchronize the client database with the latest update or configuration from the master database by comparing (functions of) identifiers and version stamps).

13. For claims 3 and 15, the claims are rejected as in claims 1 and 13. Multer-Song-Vert-Chasman-Sudharshana further discloses the associated device profile relates to at least one of a hardware profile and a software profile (Multer, fig. 11-13, Chasman, fig. 1, business software application).

14. For claims 4 and 16, the claims are rejected as in claims 2 and 14. Multer-Song-Vert-Chasman-Sudharshana further discloses initiating a process to configure the mobile device further includes receiving a device registry file from the mobile device (Multer, [0229]-[0233], pushing device application configuration from device to server, Chasman, fig. 4, step 430, update request in a queue with version stamps reads on registry information); determining device specific files appropriate for the first device as a function of the associated configuration parameter and the device registry associated with the mobile device (Chasman, fig. 4, after synchronization, based on identifiers and version registry information, determine the appropriate new version information).

15. For claims 5 and 17, the claims are rejected as in claims 1 and 13. Multer-Song-Vert-Chasman-Sudharshana further discloses each of the configuration parameters associates the application with at least one configuration option (Multer, fig. 12, Chasman, fig. 2, each application (type) has a one configuration (at least version identifier) associated with it).

16. For claims 6, 12 and 18, the claims are rejected as in claims 5, 8 and 17. Multer-Song-Vert-Chasman-Sudharshana further discloses the at least one configuration option includes at least one of power settings, menu options and application settings (Multer, fig. 12, application items settings, Chasman, fig. 3, application settings of a business object).

17. For claims 7, 11 and 19, the claims are rejected as in claims 1, 8 and 13. Multer-Song-Vert-Chasman-Sudharshana further discloses the at least one mobile device includes at least one of a laptop computer and a PDA ("Personal Digital Assistant") (Chasman, [0017] lines 1-7).

18. For claim 20, Multer discloses a method for deploying application resources to a mobile device comprising:

receiving at a server a synchronization request from a mobile device running an application ([0222], sync triggering from a device by a particular application), the synchronization request to include a device identification (ID) ([0223], device name ID);

determining a device profile as a function of the device ID ([0223], user device profile by device ID);

retrieving a configuration parameter associated with the application (fig. 12, [0241], hierarchy containing application relations with associated items or configuration parameters);

determine whether application resources corresponding to a resource ID are already installed on the mobile device (fig. 12, [0220], determine whether synchronization of application items is necessary);

using the resource ID to retrieve the application resources from an application resource database; and initiating a process to install the application resources on the mobile device (fig. 15, [0223], determine device profile, then [0225]-[0227], resolving conflicts in data by finding data deltas and applying the data deltas to appropriate application items).

Multer does not disclose storing, at the server, in a descriptor file a registry for each of the at least one mobile device; and querying, in the descriptor file, a registry associated with the mobile device.

However, Song discloses storing and accessing a database of registry of each mobile device (fig. 3, 4, each device has a registry record at the registry server)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer and Song to maintain a device specific registry record at the server so that device specific information can be retrieved separately.

Multer-Song does not disclose each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device;

However, Vert discloses each registry maintaining application resources and configuration parameters relating to each software application that are currently installed on the each of the at least one mobile device (fig. 4, a registry for a machine with installed programs 1-n and associated program configuration parameters such as name and data fields on the right)

It would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song and Vert to maintain a device specific registry record associated with all installed applications at the server so that device specific application information can be retrieved separately.

Multer-Song-Vert does not disclose the server is a middleware server.

However, Chasman discloses the same (fig. 1, fig. 2, application server with a master database for storing business object type or application and configuration information, application server is read as an middleware server, see Yu et al. US 2007/0226155, [0100], an application server is a middleware server)

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song, Vert and Chasman to implement the synchronization scheme of Multer to a dedicated server such as a middleware server to improve system performance.

Multer-Song-Vert-Chasman does not disclose: generating a resource ID from both the device profile and the configuration parameter

However, Sudharshana discloses: associating a resource ID with both the device profile and the configuration parameter (3.9, 3.11, generating a resource patch ID using device ID and configuration parameter—software version; 3.13, generated patch download from management server to mobile device)

Therefore, it would have been obvious for one skilled in the art at the time of the invention to combine the teachings of Multer, Song, Vert and Chasman and Sudharshana to implement simplify management of patches or resources to be deployed to mobile devices by generating ID's for each patch.

Multer-Song-Vert-Chasman-Sudharshana does not disclose generating a resource ID from both the associated device profile and the associated configuration parameter. Sudharshana teaches generating a unique ID of a resource (patch) associated device profile and the associated configuration parameter (3.11 first par., ME model and e.g., software version).

However, Noble discloses a method of generating a unique ID based on some parameters ([0009], claim 6, generating a unique ID based on a given network ID and a device ID)

It would have been obvious to one skilled in the art at the time of the invention to combine the teachings of Multer-Song-Vert-Chasman-Sudharshana and Noble to generate a resource ID uniquely from given parameters such as device profile (model) and associated configuration parameter (software version). The motivation would be to uniquely distinguish each generated ID with one another efficiently and to backwardly retrieve the parameters from the ID if needed.

Conclusion

19. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure is included in form PTO 392.

20. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hieu T. Hoang whose telephone number is 571-270-1253. The examiner can normally be reached on Monday-Thursday, 8 a.m.-5 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thu Nguyen can be reached on 571-272-6967. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/HH/

/THU NGUYEN/
Supervisory Patent Examiner, Art Unit 2452